Claim(s)

[0115] Having thus described the preferred embodiments, the invention is now claimed to be:

1. A method of designing hardware, said method comprising:

(a) entering source code into a source code file, said source code using a context-free grammar that describes a job the hardware being designed has to do rather than describing an implementation of the hardware; and,

(b) compiling the source code file to generate an output file which describes an optimized state machine for implementing the hardware, said output file being written in a C-based code.

2. The method according to claim 1, wherein from the same source code file, output files can be generated in step (b) which describe both transmit and receive state machines.

3. The method according to claim 1, further comprising:

(c) converting the output file to at least one of a hardware description language and a register transfer level code.

4. The method according to claim 1, wherein optimization of the state machine is achieved by:

identifying sequences of states in the source code file that occur more than twice in a row;

collapsing the identified sequence into one state in the output file; and,

wrapping a counter around the one state.

5. The method according to claim 1, wherein a BNF style notation is used to create the source code.

6. Residing on a computer-readable medium, computer software for designing hardware, said software comprising:

a compiler which generates from source code an output file that describes an optimized state machine for implementing hardware, said source code using a context-free grammar that describes a job that the hardware being designed has to do, and said output file being written in a C-based language.

7. The software according to claim 6, wherein from the same source code, the compiler can generate output files which describe both transmit and receive state machines.

8. The software according to claim 6, wherein optimization of the state machine is achieved by the compiler identifying sequences of states in the source code that occur two or more times in a row; collapsing the identified sequence in the output file such that the sequence occurs only once; and, wrapping a counter around the collapsed sequence.

9. The software according to claim 6, wherein said software further comprises:

a node tracking option such that when a user selects the node tracking option, the compiler puts variables in the output file which have names matching those of corresponding nodes in the source code, said variables getting set to a defined value in every state that is generated from its corresponding node.

10. A system for designing hardware, said system comprising:

a computer having means for entering an input file written in a source code, said source code using a context-free grammar which describes a job that hardware being designed has to do; and,

a compiler which runs on the computer, said compiler selective converting the input file into an output file which is written in a C-based code, said output file describing an optimized state machine for implementing the hardware being designed.

11. The system according to claim 10, wherein the system further comprises:

means for translating the output file from the C-based code to a hardware description language.

12. The system according to claim 11, wherein the hardware description language is one of VDHL and Verilog.

13. The system according to claim 10, wherein from the same input file, the compiler can selectively generate output files which describe both transmit and receive state machines.

14. The system according to claim 10, wherein optimization of the state machine is achieved by:

identifying sequences of states in the input file that repeat;

collapsing the identified sequence into a single state in the output file; and,

wrapping a counter around the single state.

15. The system according to claim 10, wherein a BNF style notation is used for the source code.